

CLAIMS

What is claimed is:

1. A computer-implemented method comprising:
 - detecting a change introduced into a software object of a first software subsystem, wherein the software object is used by software objects of a second software subsystem;
 - allowing the change if the change is compatible with the objects of the second software subsystem without introducing any changes into the software objects of the second software subsystem.
2. The method of claim 1 further comprising determining whether the change is predefined as compatible.
3. The method of claim 2 further comprising allowing the change if the change is predefined as compatible.
4. The method of claim 3 further comprising issuing a message that the change is not allowed if the change is not predefined as compatible.
5. The method of claim 4 further comprising allowing the change if an expert declares the change compatible upon receiving a request for a manual compatibility check, wherein the change is not predefined as compatible.
6. A computer-implemented method comprising:

identifying a subset of software objects of a first software subsystem
and declaring the subset of software objects frozen;

detecting a change introduced into a frozen software object from the
subset of software objects, and prior to allowing the change determining
whether the change is compatible with a second software subsystem.

7. The method of claim 6 wherein the subset of software objects declared frozen includes software objects of the first software subsystem that are used by the second software subsystem.
8. The method of claim 7 wherein frozen objects are classified to include released objects and restricted objects.
9. The method of claim 8 wherein the released objects include objects that are used by the second software subsystem without restrictions.
10. The method of claim 8 wherein the restricted objects include objects that are used by a small number of objects of the second software subsystem.
11. The method of claim 8 wherein an identification of recent changes introduced into a restricted object is provided when objects of the second software subsystem request new usage of the restricted object.
12. The method of claim 8 wherein classification of the frozen objects is based on a number of times a frozen object is used by the second software subsystem.

13. The method of claim 6 wherein a software object is a function module.
14. The method of claim 6 wherein a software object is a data structure.
15. The method of claim 13 wherein the software object includes an environment of the function module.
16. The method of claim 6 wherein a software object includes a class and an environment of the class.
17. The method of claim 6 wherein a software object includes an interface and an environment of the interface.
18. The method of claim 6 wherein a software object includes a program and an environment of the program.
19. The method of claim 6 wherein the detecting the change comprises automatically monitoring development of software code.
20. The method of claim 6 wherein the determining whether the change is compatible comprises determining whether there is a predefined declaration of compatibility of the change.

21. The method of claim 7 wherein the determining whether the change is compatible comprises determining whether an expert declared the change compatible.
22. A computer-implemented method comprising:
- performing a global compatibility check of software objects of a first software subsystem by determining whether any changes were introduced into a subset of the software objects of the first software subsystem since the time of a last compatibility check, wherein the introduced changes were introduced without obtaining prior approval;
 - identifying software objects of a second software subsystem affected by an unapproved change, wherein the affected software objects of the second software system are software objects using at least one software object of the subset of the software objects of the first software system; and
 - issuing a notice of possible incompatibility between affected software objects and software objects including the unapproved change.
23. The computer-implemented method of claim 22 wherein the performing a global compatibility check comprises comparing a current version of software code with a version of the software code at a time of a last global compatibility check.
24. The method of claim 22 wherein the subset of the software objects includes frozen software objects.

25. The method of claim 24 wherein the frozen software objects include objects of the first software subsystem used by objects of the second software subsystem.

26. An apparatus comprising:

a changes monitor to automatically detect a change introduced into a software object of a first software subsystem, wherein the software object is used by objects of a second software subsystem, and the changes monitor to allow the change if the change is compatible with the objects of the second software subsystem without introducing any changes into the objects of the second software subsystem; and

an error notification module to notify a software developer introducing the change into the object of the first software subsystem of a not allowed change if the change is incompatible.

27. The apparatus of claim 26 wherein the changes monitor to allow the change if the change is compatible comprises the changes monitor to determine whether there is a predefined declaration of compatibility of the change.

28. The apparatus of claim 26 wherein the change monitor to allow the change if the change is compatible comprises the changes monitor to determine if an expert declares the change compatible upon receiving a request for a manual compatibility check, wherein the change is not predefined as compatible.

29. The apparatus of claim 26 further comprising a master system including the changes monitor to detect a change introduced into a software object of a first software subsystem from a plurality of software subsystems.

30. The apparatus of claim 26 wherein the first software subsystem is located at a server.

31. The apparatus of claim 26 wherein the second software subsystem is located at a client.

32. An article of manufacture comprising:

a storage medium having stored therein instructions which, when executed by a processor, cause a processing system to perform a method comprising:

detecting a change introduced into a software object of a first software subsystem, wherein the software object is used by software objects of a second software subsystem;

allowing the change if the change is compatible with the objects of the second software subsystem without introducing any changes into the software objects of the second software subsystem.

33. The article of manufacture of claim 32 wherein the instructions, which when executed by the processor, cause the processing system to perform the method further comprising determining whether the change is predefined as compatible.

34. The article of manufacture of claim 32 wherein the instructions, which when executed by the processor, cause the processing system to perform the method wherein the method further comprising issuing a notification that the change is not allowed if the change is not predefined as compatible.

35. The article of manufacture of claim 32 wherein the instructions, which when executed by the processor, cause the processing system to perform the method wherein the method further comprising allowing the change if an expert declares the change compatible upon receiving a request for a manual compatibility check, wherein the change is not predefined as compatible.

36. An apparatus comprising:

means for detecting a change introduced into a software object of a first software subsystem, wherein the software object is used by software objects of a second software subsystem;

means for allowing the change if the change is compatible with the objects of the second software subsystem without introducing any changes into the software objects of the second software subsystem; and

means for issuing a notice of a not allowed change if the change is not compatible.

37. The apparatus of claim 36 further comprising means for allowing the change further comprise means for determining whether the change is predefined as compatible.

38. The apparatus of claim 36 wherein means for allowing the change further comprise means for allowing the change if an expert declares the change compatible upon receiving a request for a manual compatibility check, wherein the change is not predefined as compatible.